

## Web Audio Evaluation Tool - Bug #1624

### In the AB interface, comment boxes are added but not removed

2016-02-22 06:24 PM - Giulio Moro

<b>Status:</b>	Resolved	<b>Start date:</b>	2016-02-22
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Nicholas Jillings	<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>			

#### Description

- comments of pages after the first one are added at the bottom instead of replacing comments from the previous page, so that, e.g.: after 4 pages you have four copies of the comment boxes, each retaining the text you previously input.

probably because you never empty the div before .appendChild

#### History

##### #1 - 2016-02-22 08:19 PM - Giulio Moro

- Status changed from New to In Progress
- Assignee set to Nicholas Jillings

Can be fixed with

```
diff -r cce99fc682d8 interfaces/AB.js
--- a/interfaces/AB.js  Mon Feb 22 16:26:26 2016 +0000
+++ b/interfaces/AB.js  Mon Feb 22 20:18:18 2016 +0000
@@ -152,8 +152,14 @@
     interfaceContext.comparator = new comparator(audioHolderObject);
     if (audioHolderObject.showElementComments)
     {
-    var commentHolder = document.createElement('div');
-    commentHolder.id = 'commentHolder';
+    var commentHolder = document.getElementById('commentHolder'); // get the commentHolder if it exists
+    if(commentHolder === null){ //if it does not, create it
+        commentHolder = document.createElement('div');
+        commentHolder.id = 'commentHolder';
+    } else { // if it exists, empty the commentHolder from the boxes
+        // belonging to the previous page
+        commentHolder.innerHTML = ""; // not the best way, potentially causes memory leak if you are referencing the DOM nodes
+        somewhere else
+    }
     document.getElementById('testContent').appendChild(commentHolder);
     // Generate one comment box per presented page
     for (var element of audioEngineContext.audioObjects)
```

I am not pushing this to the repo yet, because I am not sure if you are referencing the content of commentHolder anywhere else, which could lead to a memory leak. Have a look at it and then feel free to merge in.

##### #2 - 2016-02-23 09:48 AM - Nicholas Jillings

The comment boxes are managed by the interface object in core.js (so we don't reinvent the wheel for each test page). Likely it is never getting called to scrub the boxes.

**#3 - 2016-02-23 09:59 AM - Giulio Moro**

look at my fix in the previous comment!

**#4 - 2016-02-23 10:01 AM - Nicholas Jillings**

Yes I know, but there are more things behind it that I'd rather fix that's all. Since it is a global object, if it is cropping up here it could crop up else where

**#5 - 2016-02-23 10:50 AM - Nicholas Jillings**

- *Status changed from In Progress to Resolved*

Fixed in dev\_main. The InterfaceContext object now explicitly removes the node from the page, then destroys the object on each page reload.